Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/IL05/000241

International filing date: 01 March 2005 (01.03.2005)

Document type: Certified copy of priority document

Document details: Country/Office: US
Number: 60/548,182
Filing date: 01 March 2004 (01.03.2004)

Date of receipt at the International Bureau: 15 March 2005 (15.03.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)

World Intellectual Property Organization (WIPO) - Geneva, Switzerland
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

January 18, 2005

THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THE UNITED STATES PATENT AND TRADEMARK OFFICE OF THOSE PAPERS OF THE BELOW IDENTIFIED PATENT APPLICATION THAT MET THE REQUIREMENTS TO BE GRANTED A FILING DATE UNDER 35 USC 111.

APPLICATION NUMBER: *60/548,182*
FILING DATE: *March 01, 2004*

By Authority of the
COMMISSIONER OF PATENTS AND TRADEMARKS

M.K. Hawkins
M. K. HAWKINS
Certifying Officer

Approved for use through 01/31/98. OMB 0651-0037
Patent and Trademark Office; US DEPARTMENT OF COMMERCE

PROVISIONAL APPLICATION FOR PATENT COVER SHEET

This is a request for filing a **PROVISIONAL APPLICATION FOR PATENT** under 37 CFR 1.53 (b)(2).

Docket Number	**27623**	Type a plus sign (+) inside this box ->	+

INVENTOR(s) / APPLICANT(s)

LAST NAME	FIRST NAME	MIDDLE INITIAL	RESIDENCE (CITY AND EITHER STATE OR FOREIGN COUNTRY)
TSAFRIR	Ilan		Rehovot, Israel
TSAFRIR	Dafna		Rehovot, Israel
DOMANY	Eytan		Rehovot, Israel

TITLE OF THE INVENTION (280 characters max)

SORTING POINTS INTO NEIGHBORHOODS (SPIN)

CORRESPONDENCE ADDRESS

G. E. EHRLICH (1995) LTD.
c/o ANTHONY CASTORINA
2001 JEFFERSON DAVIS HIGHWAY
SUITE 207

STATE	VIRGINIA	ZIP CODE	22202	COUNTRY	USA

ENCLOSED APPLICATION PARTS *(check all that apply)*

☑ Specification	*Number of Pages*	47	☑ Applicant is entitled to Small Entity Status
☑ Drawing(s)	*Number of Sheets*	11	☑ Other (specify) *ASSIGNMENT* 44 Claims

METHOD OF PAYMENT OF FILING FEES FOR THIS PROVISIONAL APPLICATION FOR PATENT *(check one)*

□ A check or money order is enclosed to cover the filing fees ☑ The Commissioner is hereby authorized to charge filing fees and credit Deposit Account Number:	50-1407	FILING FEE AMOUNT ($)	$ 80.-

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.

☑ No

□ Yes, the name of the US Government agency and the Government contract number are: ______________

Respectfully submitted,

SIGNATURE ______________________

26 February 2004
Date

25,457
REGISTRATION NO.
(if appropriate)

TYPED or PRINTED NAME SOL SHEINBEIN

□ Additional inventors are being named on separately numbered sheets attached hereto

USE ONLY FOR FILING A PROVISIONAL APPLICATION FOR PATENT

Burden House Statement: This form is estimated to take 2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Box Provisional Application, Assistant Commissioner for Patents, Washington, DC 20231.

US PROVISIONAL APPLICATION

Title: Sorting Points Into Neighborhoods (SPIN)

Inventors: Ilan Tsafrir, Dafna Tsafrir, and Eytan Domany

FIELD OF THE INVENTION

The present invention is of a method for analyzing and visualizing large collections of data.

BACKGROUND OF THE INVENTION

Exploratory data analysis is critical in a broad range of research areas, where large collections of data need to be meaningfully arranged and presented. Indeed, a major challenge in the analysis of large-scale multidimensional data is effective organization and visualization. Graphically structured presentation can greatly aid humans in data mining: a clear and interactive display may reveal subtle structure and relationships, and assist in tracking down elusive connections.

SUMMARY OF THE INVENTION

The background art does not teach or suggest an efficient, intuitive tool for automated analysis and visualization, which may optionally be performed with little or no manual intervention. The background art also does not teach or suggest reorganization of distance matrices using the characteristics of the distances themselves. The background art does not teach how to read the properties and relationships of the data from the reordered distance matrix.

The present invention overcomes these deficiencies of the background art by providing a method for an unsupervised analysis of data according to a reordered distance matrix. According to preferred embodiments thereof, the present invention is useful for large scale multidimensional data, more preferably data having at least four dimensions. The present invention is also preferably used for data comprising a plurality of objects characterized by continuous variables, for example variables having a continuum of possible values rather than a plurality of discrete values. It should be noted that single object featuring a plurality of points would also be considered a plurality of objects with regard to the present invention.

According to preferred embodiments, the present invention provides an analysis method termed herein *SPIN*, a novel method for the organization and visualization of data, implemented in a simple tool. *SPIN* utilizes traits of distance matrices to sort objects in a natural ordering that highlights the underlying structure of the original, multidimensional data. The shape of the distribution of objects and/or of the objects themselves, and relationships between objects can be inferred from the reordered distance matrix generated by *SPIN*. As an unsupervised analysis tool, *SPIN* does not rely on any external labels, but rather explores the inherent characteristics of the data. In the analysis of high-throughput biological experiments, discretely-labeled data, such as clinical labels of 'sick' versus 'healthy', is traditionally organized by various clustering approaches. However, when the objects are characterized by continuous variables, e.g. survival intervals of patients or expression levels of genes, any sharp separation into distinct clusters will be rather arbitrary. Thus, a different organization approach, one which emphasizes ordering rather than grouping, could be more relevant.

This work focuses on finding a one-dimensional ordering of a set composed of n data points, and to present as output the matching (2-dimensional) n by n distance matrix D. An element D_{ij} of D represents the dissimilarity between objects i and j. Our aim is to find a permutation of the data points, such that the correspondingly reordered distance matrix reveals the underlying structure of the data, utilizing the human ability to readily recognize patterns in color images [1]. Sorting Points Into Neighborhoods (*SPIN*), generates a one-dimensional ordering of the objects and presents the reordered distance matrix in an intuitive color coded image that allows the observer to infer the underlying structure of the data. *SPIN* is especially suitable for analyzing high-throughput biological experiments, such as gene array experiments, where results are typically summarized in an expression matrix, in which each element denotes the expression level of a particular gene in a specific sample [1]. In this context two types of distance matrices can be produced: the distances between all pairs of samples can be calculated based on their expression levels over the measured genes, and the distance between all pairs of genes can be measured in the sample dimensions [2]. The sorted distance matrix generated by *SPIN* is particularly useful in time-series experiments, where an elongated cluster represents the temporal evolution of a particular biological module, such as cell-cycle progression. Another example where the shape revealed by *SPIN* has a clear biological interpretation comes from cancer research where samples are often composed of mixtures of cells: for instance, colon tissue samples isolated from liver metastases arrayed into an elongated, ellipsoid cluster [3]. The genes that induced the elongation were characteristic of liver, suggesting that this pattern reflects a mixture of the metastasis samples with cells originating from the liver.

Among the many advantages of the present invention is that the method provides an efficient and intuitive way to read the properties and relationships of the data from the reordered distance matrix. Contact maps of proteins have been used to discover secondary structure, but they posses an inherent ordering (according to the primary sequence). Therefore, the present invention represents the first method to be able to discover such properties and relationships without any inherent ordering (that is to say, pre-ordering) of the data.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention herein described, by way of example only, with reference to the accompanying drawings, wherein:

FIG 1 shows an exemplary analysis of a set of points that form a single object in multidimensional space;

FIG. 2 shows analysis of a data set composed of several distinct clusters;

FIG. 3 illustrates *SPIN*'s ability to deal with complex objects embedded in high dimensional space;

FIG. 4 shows a schematic illustration of the side-by-side algorithm according to the present invention;

FIG. 5 is an exemplary pseudocode of an exemplary side-by-side algorithm according to the present invention;

FIG. 6 shows the end result of applying Side-to-side to data composed of 960 points in 9 spherical clusters in 3D;

FIG. 7 is an exemplary pseudocode of an exemplary neighborhood algorithm according to the present invention;

FIG. 8 shows a comparison between side-by-side and neighborhood algorithms;

FIG. 9 shows the results of analyzing yeast data with the method according to the present invention;

FIG. 10 shows the results of analyzing cancer data with the method according to the present invention; and

FIG. 11 shows the results of using the method according to the present invention for machine vision.

DESCRIPTION OF PREFERRED EMBODIMENTS

The present invention is of a method for an unsupervised analysis of data according to a reordered distance matrix. According to preferred embodiments thereof, the present invention is useful for large scale multidimensional data, more preferably data having at least four dimensions. The present invention is also preferably used for data comprising a plurality of objects characterized by continuous variables, for example variables having a continuum of possible values rather than a plurality of discrete values.

According to preferred embodiments, the present invention provides an analysis method termed herein *SPIN*, a novel method for the organization and visualization of data, implemented in a simple tool.

The input to SPIN is a distance matrix, and its output is a reordered distance matrix, obtained by permuting the *N* objects. Currently two different algorithms, based on two complementary intuitions, are implemented. However, optionally and preferably substantially any algorithm may be employed with the method of the present invention. The two algorithms utilize two distinct (and sometimes competing) desirable properties of properly ordered distance matrices: *first*, in many cases the values in the upper rows of a well-ordered distance matrix tend to increase with the column index, while the values in the bottom rows have the opposite inclination. In other words, the slope of the linear regression of the values in a row is a decreasing function of the row's index in the sorted matrix. The first algorithm, named *Side-to-side*, simply generates such a pattern. The second property is that the region near the main diagonal tends to have smaller dissimilarity values, i.e. a "good" ordering locates points next to their neighbors in the full high-dimensional space. The *second* algorithm, called *neighborhood,* tries to create such an arrangement by ensuring that distant data points in the multi-dimensional space are not placed close to each other in the linear ordering.

Although both algorithms achieve a one dimensional ordering of the data set, the final resulting permutations are different in the following sense: Side-to-side tries to capture a particular pattern in the image of the distance matrix. As a result, points that are placed far apart in the linear ordering are also distant in the full high-dimensional space. Neighborhood, on the other hand, tries to make sure that neighboring points in the linear ordering are close to each other in the high dimensional space. This subtle distinction in emphasis may lead to substantial difference in the results, as illustrated for points that form a ring, described in greater detail below. A ring is a simple example where these two criteria are mutually exclusive. Neighborhood orders the points around the circumference of the ring. Due to the cyclic symmetry of the ring, the end points in this ordering are very close to one another in the true high dimensional space. This does not conform to the pattern that Side-to-side enforces.

In general, Side-to-side is simpler, faster, and seems to converge quickly for all the examples currently examined. It has no parameters, so the final ordering depends only on the initial permutation. Neighborhood, on the other hand, seems to have the potential to generate superior results in several cases. However, it does not always converge, and occasionally gets mired in obviously local stationary permutations. The size of the neighborhood, σ, determines the typical scales of objects that are revealed: small values of σ tend to break large clusters into small "clumps"; conversely, large σ values tend to merge neighboring clusters.

Several examples are given below in which the structure uncovered by *SPIN* has a clear biological interpretation, such as the cyclic nature of cell-cycle progression, visualized in a ring conformation. In another example the tissue composition of tested samples is captured by their relative placement in an ordered elongated cluster, formed in the space of tissue specific genes. Another example is related to machine or robot vision. Therefore, the method of the present invention has general applicability, which makes it relevant to diverse scientific disciplines and/or technologies.

Example 1 demonstrates the concepts and the intuitions that underlie the method according to the present invention, and shows how to infer structure characteristics from the sorted distance matrix. Example 2 provides a formal description of a preferred illustrative embodiment of the method. Examples 3-4 feature several applications to real data, where the shapes uncovered by *SPIN* are directly interpretable in biological terms. Example 5 relates to data for machine or robot vision.

Section I: General Description of an Illustrative Method

Example 1

Pedagogical examples

A properly ordered distance matrix is indicative of the shape of a set of points. All the data sets presented in this article were ordered using *SPIN*, starting from a random initial permutation. The distance matrices were generated using the Euclidean distance measure, though our methodology can be applied to many dissimilarity metrics. The color of element D_{ij} reflects the relative distance between points i and j, where blue (red) denotes small (large) distances, respectively.

For explaining the *SPIN* method, we first address a set of points that form a single object in multidimensional space. The top row (1) of fig. 1 depicts the placement of $n=500$ points in $d=3$ dimensions, for a few toy data sets; below each object (row 2) we show the initial, unordered, distance matrix, while in the bottom row we present the corresponding sorted distance matrix. Although both the ordered and unordered matrices contain exactly the same elements, the sorted distance matrix allows a human observer to deduce structural information. The colors of the objects in the top row represent the linear ordering of the points, where the first point is dark blue ranging to the last point in dark red. This is the same order that *SPIN* imposed on the distance matrix, i.e. the first row and first column contain the distances from the first point (the one colored dark blue in the PCA image) to all other points. (a) An elongated shape, in this case a cylinder, displays a clear gradient of distances that increase as one moves away from the main diagonal. (b) This gradient holds even if the center-line of the elongated shape is curved, as demonstrated by a helical structure. (c) A ring of points is characterized by a cyclic pattern, with small distances (blue) at the corners. (d) Finally, a spherical cluster with no significant elongation has a diffuse texture. As a rule, the smoothness of the texture in the image of the distance matrix is a function of the elongation of the cluster.

For example, consider points uniformly distributed within a cylinder, as presented in fig. 1a1. The first stage in our methodology is to generate a distance matrix for a set of points. The initial, unordered, distance matrix (fig. 1a2) is not easy to interpret, but after rearrangement in *SPIN* the sorted image is highly informative (fig. 1a3). In this example *SPIN* orders the points from one end of the cylinder to the other, so that the correspondingly reorganized distance matrix has a characteristic pattern. The area close to the main diagonal of the sorted matrix contains only short distances (blue color), with a clear gradient of increasing distances (colors vary from blues to reds) as one moves away from the main diagonal. In fact, this signature characterizes any significantly elongated object, as can be seen in the case of a coil (fig. 1b). This colored pattern is the essence of *SPIN*: neighboring points in the one dimensional ordering produced by *SPIN* are also close to each other in the original multidimensional space. Hence, once a distance matrix is sorted in

SPIN, simply viewing the resulting colored image allows the user to deduce the object's features.

Another simple structure, a ring (see fig. 1c1), is also characterized by a definitive pattern, once the points are properly ordered. In the sorted image (fig. 1c3) the blue region around the main diagonal indicates that *SPIN* sorts the points around the circumference of the ring, placing points that are close in the original space as neighbors. The distances in the sorted matrix are cyclic with regard to their position relative to the main diagonal. This can be understood by considering the organization of the ring: starting from any arbitrary point and going around the ring, the distance of the current point to the initial point increases monotonously (colors change from blue to red) until the diametrically opposing point is reached. At this stage the distances begin to decrease (colors go back to blue), as we approach the point of origin from the other side.

Given more complex data, the ordered distance matrix suggested by *SPIN* can capture the over all layout of a compound structure, as well as the local conformation of various components. In fig. 2 we analyze a data set composed of several distinct clusters. The sorted distance matrix that *SPIN* produces allows one to study the local shape of each cluster in the data, as well as the global relationships between clusters. In this example there are four major clusters, two of which are tight spherical clusters (eyes) that appear as dark blue squares on the main diagonal. From the light blue color of the squares between them we can deduce that the eyes are relatively close to each other, i.e. we can infer their relative placement. The next cluster (smile) has a gradient of colors, from dark blue on the main diagonal to light blue at the corners. As explained above, this indicates an elongated structure. The fourth cluster has a sharp gradient of colors, that cycles through the entire spectrum. As explained in fig. *1*, such a pattern in the distance matrix indicates a cyclic shape, in this case a ring. The fact that the distance between opposing points on the ring is the largest in the data set (i.e. the darkest red in the distance matrix) indicates that the ring engulfs all other points.

This toy data set is composed of *800* points in 10-dimensions. The complex object was originally generated in 3-D, and then seven additional dimensions of noise (uniformly distributed between -1 and 1) were added. The right image of fig.2 shows the projection of the points onto the first and second PCA plane (two spheres and a curved cylinder within a ring), and the distance matrix on the left is sorted accordingly. From this organized matrix one can easily infer the shapes of the four clusters and their relative placement. For example the position on the ring closest to the top eye is denoted by a black circle. This can be inferred from the sorted matrix by locating the blue patch in the region corresponding to the relationship between the eye and the ring, as shown by the arrows.

The next example illustrates *SPIN*'s ability to deal with complex objects embedded in high dimensional space: Figure 3a1 shows a 3-D projection of points constituting a set of seven intersecting cylinders, twisted in $d=7$ dimensions. On the left is a set of seven orthogonal intersecting cylinders, comprised of *1400* points in seven dimensions. The rods were twisted by rotation with angles that increase linearly with the distance from the origin. (a1) The points displayed in the first three *PCA*, colored according to their placement in *SPIN*. In this example the coloring is crucial for making sense out of a complex image. (a2) The correspondingly sorted distance matrix is shown. The right column shows a simplified version having *600* points composing three straight intersecting rods in three dimensions. (b1) The actual placement of the points is shown, where the numbered arrows illustrate the order imposed by *SPIN*. (b2) The correspondingly sorted distance matrix is shown. The region of the intersection creates blue patches in the off-diagonal regions of the distance matrix (denoted by *a*).

In the distance matrix in fig. 3a2 each rod is an elongated structure along the main diagonal. As explained above, the relationships between the seven regions can be deduced from the shape of the off-diagonal regions in the organized distance matrix, and indeed the fact that the rods share a common nexus is reflected by a grid of blue patches. This example illustrates a case where SPIN highlights a simple characteristic of a high-dimensional object that is not immediately made evident by projection onto a smaller dimensional space. In order to assist the reader in understanding this example, the right column in fig. 3 is a simplified version: three straight rods in three-dimensions. The arrows follow the order of the points, going from the outer edge of a rod, through the center then out along a second rod, jumping to the outer edge of the third rod etc.

Example 2

Illustrative Method

This Example provides an illustrative method according to the present invention, as a description of a preferred embodiment thereof, the *SPIN* method.

The input to *SPIN* is a distance matrix $D_{n\times n}$ calculated for a data set composed of n points, and its output is a reordered distance matrix, obtained by permuting the n objects according to a particular permutation $P \in S_n$ (the permutation group of n points). We denote by P also the permutation matrix associated with p.

In order to find criteria for a good ordering, we studied several simple objects characterized by an inherent natural ordering (See fig. 1a-c). Having observed such ordered distance matrices, we noticed two distinct and sometimes competing properties. First, in many cases the values in the upper rows of a well-ordered distance matrix tend to increase with the column index, while the values in the bottom rows have the opposite inclination. The second property is that the region near the main diagonal tends to have smaller dissimilarity values, i.e. points are positioned next to their neighbors in the full high-dimensional space. These two properties 'Side-to-Side' and 'Neighborhood', respectively, and are related to two algorithms of the same name.

These attributes can be mathematically formulated by introducing an energy function $F \equiv F_D : S_n \rightarrow \Re$ quantifying the fitness of every matrix ordering. Thus, the ordering problem becomes finding the permutation p minimizing F. We emphasize that there is no unique 'correct' choice of F, as different energy functions may potentially reveal different aspects of the data, thus enabling study of diverse properties, as will be demonstrated later.

The two aforementioned desired features of an ordered distance matrix can be represented by the following energy functions:

1. *Side-to-Side (STS)*: Let X be a strictly increasing (column) vector. Set $F(P) = X^T PDP^T X$.

2. *Neighborhood*: Let W be a symmetric weight matrix concentrated in a region, determined by a parameter σ, around its main diagonal. Set $F(P) = tr(PDP^T W) = \sum_{i,j=1}^{n} W_{ij} D_{P(i)P(j)}$, where *tr* denotes the matrix trace.

Interestingly, the problems of minimizing the two choices of F mentioned above are special cases of a more general optimization problem, known as the *Quadratic Assignment Problem* (*QAP*), introduced by [4]. The *QAP* formulation is as follows: Given two $n \times n$ matrices D and W: find $P \in S_n$ that minimizes $tr(PDP^T W)$. Note that $W = XX^T$ corresponds to the *STS* problem.

The general *QAP* is considered an extremely difficult optimization problem. It is known to be *NP-Hard* even to approximate, and in practice, usually untractable for n more than *30* (See [5] for a comprehensive survey of the problem). The particular choices of F that were made for the present

Examples are shown to be also NP-hard, and therefore two analogous heuristic search algorithms were proposed, aimed at finding a global minimum.

These two algorithms are now explained in more detail below.

Side-to-side.

The algorithm of *STS* is summarized as follows:

```
Input: $D_{nxn}$ and a strictly increasing vector $X$
1. Compute $S = D X$.
2. Sort $S$ in descending order to get $S' = P(S)$, where $P$ is the sorting permutation.
3. If $P(S) != S$, set $D = P D P^T$ and go to $1$.
4. Output $D$.
```

Given a distance matrix D, multiply it by a weight-vector W; the resulting vector S is termed "scores" (see Figure 4). Since dot product is a measure of distance between two vectors, the scores reflect the degree of similarity between every row in the input matrix and the weight-vector. Our particular choice of weights, $W_j = (2j - N - 1)/(N - 1)$, is a linearly ascending vector, from -1 to 1. Hence, the score of a particular line reflects the slope of the linear regression of its values.

In the second step the score vector is sorted in descending order, and this is taken as the new ordering of the points. Since the distance matrix is symmetrical, reordering the points dictates rearranging both rows and columns. The change in the order of the columns alters the order of the values in all rows. This means that if we repeat the process of scoring, the new score of a row will, in general, differ from the old one. This is resolved by iterating the process of scoring and sorting.

We call each time we pass steps 1-3 a *STS* iteration, whose complexity is $O(n^2)$. Each STS iteration can be viewed as a mapping from the permutation group S_n to itself, $G_D : S_n \text{->} S_n$. Thus P is a possible output of *STS* if and only if it is a fixed point of G_D. Note that the resulting fixed point may not be a global minimum of F, as for different initial permutations the algorithm may terminate at different fixed points, with different values of F. A known strategy to cope with this problem is to start the algorithm from many randomly generated initial permutations, and choose the best fixed point obtained. Moreover, it is also possible to have multiple global minima. For example, define for every permutation P its 'reverse' $\overline{P}$ by $\overline{P}(i) = P(n+1-i)$; $(i = 1, ..,n$). If X is anti-symmetric we get $F(P) = F(\overline{P})$, leading to at least two global minima. Some data sets may even contain further degeneracies due to inherent symmetries.

As a concrete example, when the algorithm is applied to data comprised of three well separated "superclusters", each of which consists of three dense spherical sub clusters close to each other (see Fig. 6), the sorted distance matrix displays clearly the structure of the data. Figure 6 shows the end result of applying Side-to-side to data composed of 960 points in 9 spherical clusters in 3D. The left image presents the final ordering of the distance matrix. The middle graph presents the final score vector. The right most image displays the data points in the first and second PCA.

The three super-clusters are visible as dark blue squares along the main diagonal and their actual separation in the true multi-dimensional space is captured by the colors of the regions connecting these dark squares. At a higher resolution, the three sub clusters are also apparent. Furthermore, their relative positions can be inferred by the shading of the relevant rectangles in the distance matrix. The sizeable separation of the super-clusters is reflected in the final score vector in the form of large jumps that correspond to the boundaries between super-clusters, and smaller jumps corresponding to individual clusters.

Neighborhood.

The algorithm of *Neighborhood* is summarized as follows:

```
Input : $D_{nxn}$ and $W_{nxn}$
1. Compute $M = D\ W$
2. Set $\mathrm{P} = \arg\min_{\mathrm{Q} \in \mathrm{S}_n} \mathrm{tr}(\mathrm{QM})$.
3. If tr $(P\ M)$ != tr $(M)$, set $D = P\ D\ PT$ and go to 1.
4. Output $D$.
```

Each passage of steps 1-3 is a *Neighborhood* iteration. Step 2 is accomplished by solving the Linear Assignment Problem. This solution reflects the best current guess for an improved location for all the data points. At every iteration, points are sent to their new location, based on the current ordering of the points. However, since all the points are permuted simultaneously, there is no guarantee that the previous assignment is optimal for the new ordering. Hence the need to re-iterate. Since the Linear Assignment Problem is known to be solvable in time $O(n^3)$ [6], the complexity of each iteration is $O(n^3)$.

This algorithm of SPIN relocates points to the local *neighborhood* that best fits them. In this context a neighborhood is defined by a positive weight matrix W_{ij} with a finite range σ. For example we use Gaussian weights, $W_{ij} = e^{\frac{(i-j)^2}{2\sigma^2}}$. The size of the neighborhood affects the scale at which objects are distinguished. By taking the product of the distance matrix with W we perform Gaussian smoothing of width σ on each of its rows; we call the result the mismatch matrix M_{ij}. The index of the minimum in the smoothed row i, termed the score S_i, reflects the best current guess for an improved location for that particular point. The vector of scores is calculated for all points i simultaneously, as explained in Figure 7. The relocation of points is achieved by sorting the score vector, same as in the first algorithm. However, more than one row can have the same index as its minimum, so tie breaks have to be accounted for. One way to do this is by using the linear assignment problem, while another is to bias the index of the minima by their actual values.

Since all the points are relocated at the same time, the points in the target regions also change, so the process of scoring and sorting is repeated iteratively, until convergence is reached or the number of iterations exceeds a preset bound.

The current implementation is as an interactive GUI so that the user chooses how to adjust σ manually. For a given data set there exists a range of relevant σ values where the resulting sorted distance matrix reflects the structure of the data at that resolution. In general, relatively large σ values correspond to working at low resolution, which allows the user to study the over all layout of the data, and observe the main separations. Smaller σ values can give a better local organization (near the main diagonal) at the expense of possibly fragmenting larger clusters. At the extreme end of small σ this is simply a nearest neighbor algorithm. One heuristic scheme that usually works well is starting with a very large neighborhood, iterating several times, then lowering ε (e.g. by a factor of 2) and so forth.

Although both algorithms find a one dimensional ordering of the data set, the characteristics of the final permutations are different in the following sense: Side-to-side (denoted *STS*) enforces a particular pattern on the image of the distance matrix, one that places red points (which denote large distances) in the top-right (and bottom-left) corners. Thus points that are placed far apart in the linear ordering are also distant in the full high-dimensional space. Neighborhood, on the other hand, tries to make sure that neighboring points in the linear ordering are close to each other in the high dimensional space. This subtle distinction in emphasis may lead to substantial difference in the results, as illustrated in Figure 8 for points in a ring formation.

The left image is the result of *STS*, which tries to position red points in the top-right (and bottom-left) corners. The image on the right is the result of neighborhood sorting, which aims to avoid placing red points near the main diagonal. As a result, the optimal Neighborhood permutation orders the points around the circumference of the ring. Due to the cyclic symmetry of the ring, the end points in this ordering are very close to one another in the original space. This does not conform to the pattern that *STS* imposes.

For both algorithms, the score is shown to be improved on every iteration, thus convergence to a fixed point is guaranteed after a finite time (see below for outline of proofs of complexity and convergence).

Proofs of Complexity

Claim : The Side-to-Side problem is NP-Hard

Proof : Let G =<V,E> be some graph on n vertices. Define D as follows:

if $(V_i, V_j) \in E$ then Dij = 1, else Dij = 2.

Set Dii = 0.

Let $k \in [1,n]$ be some integer, and set Xi = 1(i>=n-k+1). It can be easily shown that G has a clique of size k if and only if $\min_{P \in S_n} X^T P D P^T X = (k-1)k$. Thus, STS reduces to the k-clique problem, which is known to be NP-complete (Garey and Johnson [14]).

Claim : The Neighborhood problem is NP-Hard

Proof : Setting Wij = 1{|i-j|=1} gives tr(PDPTW) = $\sum_{i=1}^{n-1} D_{P(i+1),P(i)}$. We get a reduction from the Traveling Salesman Problem, known to be NP-Hard, even in the Euclidian case (Papadimitriou [15]).

Proofs of Convergence

We first give the STS algorithm in a slightly revised manner, where P operates on X instead of D :

```
Input : $D$ and $X$.
1. Set $X^0 = X$, $t = 0$, $Q = I_{n\times n}$, define $P^{-1} = I_{n\times n}$.
2. Calculate $S^t = DX^t$.
3. Find $P^t$ which sorts $S^t$ in a descending order.
4. If $P^tS^t \neq P^{t-1}S^t$, set $X^{t+1} = P^{t^T}X^0$, $Q = QP^t$, set $t = t+1$ and go to 2.
5. Output $QDQ^T$.
```

It can be easily seen that this algorithm presentation is equivalent. We now prove the following lemma :

Lemma

1. $X^{t+1^T}DX^t \le (QX^0)^T DX^t$, $\forall Q \in S_n$, $t \ge 0$
2. $X^{t+1} \neq X^t \Rightarrow X^{t+1^T}DX^t < X^{t^T}DX^t$.

Proof

1. Note that :

$$X^{t+1^T}DX^t = (P^{t^T}X^0)^T DX^t = X^{0^T}P^tDX^t \equiv X^{0^T}U^t$$

$$(QX^0)^T DX^t = X^{0^T}Q^T DX^t = X^{0^T}Q^T(P^t)^{-1}P^tDX^t \equiv X^{Q^T}U^t \qquad (1)$$

And X^Q is some permutation of X^0, for any $Q \in S_n$. But, U^t is a non-increasing vector, while X^0 is a strictly increasing vector. Thus, according to a theorem by *Hardy, LittleWood* and *Polya* (Hardy et al. [10]) $X^0U^t \le Q(X^0)U^t$, $\forall Q \in S_n$, so $X^0U^t \le X^QU^t$, as desired.

2. From the first part it follows that $X^{t+1^T}DX^t \le X^{t^T}DX^t$. Assume negatively that equality holds. Then, we get : $X^{0^T}P^{t-1}DX^t = X^{t^T}DX^t = X^{t+1^T}DX^t = X^{0^T}P^tDX^t$.

   But X^0 is increasing, P^tDX^t is decreasing and $P^{t-1}DX^t$ is a permutation of it. Therefore $P^{t-1}DX^t = P^tDX^t$, and thus $P^{t-1}DX^t$ is non-increasing, and since we have started from it, we get $P^t = P^{t-1}$ and $X^{t+1} = X^t$, which is a contradiction. ■

We can now prove the following theorem :

Theorem :

Take n distinct points $x^1, \dots, x^n \in R^d$, for some $d \in N$, and a real $p \in (1,2]$, such that :

$$D_{i,j} = \|x^i - x^j\|_p,\ 1 \le i,j \le n$$

Suppose that X is strictly monotonically increasing ($X_i < X_j \iff i < j$). Then algorithm $Side-to-Side$ converges to a point after a finite number of iterations.

Proof :

According to Thm. 2.11 in Baxter [4], D is Almost Negative Definite. That is, we have for any vector V such that $\sum_{i=1}^{n} V_i = 0$, $V^T DV \leq 0$. Since for every t X^t is a permutation of X it follows that

$$(X^{t+1} - X^t)^T D(X^{t+1} - X^t) \leq 0 \tag{2}$$

Since D is symmetric it follows that

$$\frac{X^{t+1^T} DX^{t+1} + X^{t^T} DX^t}{2} \leq X^{t+1^T} DX^t \tag{3}$$

Subtracting $X^{t^T} DX^t$ from both sides of 3 one obtains:

$$\frac{X^{t+1^T} DX^{t+1} - X^{t^T} DX^t}{2} \leq X^{t+1^T} DX^t - X^{t^T} DX^t. \tag{4}$$

But the algorithm never stays at the same point for more than one iteration (step 4), namely $X^{t+1} \neq X^t$ and therefore, according to the previous lemma:

$$X^{t+1^T} DX^{t+1} - X^{t^T} DX^t < 0$$

To conclude, the energy function $\mathcal{F}(t) = X^{t^T} DX^t$ is a strictly decreasing function of t. Therefore the algorithm terminates after a finite number of steps.

The proof from above proves that for L_p norms with $p \in [1, 2]$, $SPIN$ converges to a local minima of the 'dynamical energy' : $\mathcal{F}(X^t, X^{t+1}) = X^{t^T} DX^{t+1}$. Convergence to a global minima of $\mathcal{F}_X$ is not guaranteed. For other norms, $SPIN$ might converge to a cycle, however the cycle can be viewed as a 'local minima', since it still minimizes $\mathcal{F}(X^t, X^{t+1})$ (All the cycle has the same $\mathcal{F}$.)

Convergence of *Neighborhood* :

To prove convergence, we revise the *Neighborhood* algorithm as follows:

Input : D and W

1. Set $W^0 = W$, $P^{-1} = I_{n \times n}$, $t = 0$
2. Compute $M^t = DW^t$.
3. Set.

$$P^t = argmin_{Q \in S_n} tr(QM)$$

4. If $P^t \neq P^{t-1}$, set $W^{t+1} = P^{t^T} W$ and go back to step 2.
5. Output $P^t D P^{t^T}$.

Claim :

$$tr(P^{t+1} D P^{t^T} W) \leq tr(P^t D P^{t-1^T} W) \tag{5}$$

Proof :

$$tr(P^{t+1} D P^{t^T} W) = tr(P^{t+1} DW^{t+1}) \leq tr(QDW^{t+1})\ \forall Q \in S_n \tag{6}$$

Using the symmetry of W and the property $tr(AB) = tr(BA)$ we get :

$$\begin{aligned} tr(QDW^{t+1}) = tr(QDP^{t^T} W) = tr((QDP^{t^T} W)^T) = \\ tr(WP^t DQ^T) = tr(P^t DQ^T W) \end{aligned} \tag{7}$$

Taking $Q = P^{t-1}$ in 7 gives the desired result. ■

Using the above claim, a proof of the algorithm termination can be obtained, similarly to SPS. We skip the details here.

Proof of Neighborhood Convergence

First we revise the algorithm to an equivalent form :

Neighborhood (Rev.)

Input : Dnxn and Wnxn

1. Set W0 = W, P-1 = Inxn, t = 0.

2. Compute Mt = DWt.

3. Set $P^t = \arg\min_{Q \in S_n} \mathrm{tr}(QM^t)$.

4. If tr(PtMt) 6 != tr(Pt-1Mt-1), set Wt+1 = PtTW, t = t + 1 and go to 2.

5. Output PtDPtT .

Claim : tr(Pt+1DPtTW) <= tr(PtDPt-1TW)

Proof : tr(Pt+1DPtTW) = tr(Pt+1DWt+1) · tr(QDWt+1) $\forall Q \in S_n$

Using the symmetry of W and the property tr(AB) = tr(BA) we get :

tr(QDWt+1) = tr(QDPtTW) = tr((QDPtTW)T) =

tr(WPtDQT) = tr(PtDQTW)

Taking Q = Pt-1 gives the desired result.

According to step 4, the algorithm terminates unless a strict inequality holds in the above claim. This prevents cycles of constant energy. Since the permutation space is finite, termination in a fixed point after a finite number of steps is guaranteed.

The current implementation of *SPIN* is as an interactive GUI, which enables the user to use either *STS* or *Neighborhood*. In general, *STS* is simpler, faster, and convergence seems to be quick for all the examples we have tried so far. It has no parameters, so the final ordering depends only on the initial permutation. *Neighborhood*, on the other hand, seems to capture features of the data which are missed by *STS*. For *STS*, one exemplary choice of weights is

$X_j = \frac{2j-n-1}{n-1}$, which is an anti-symmetric, linearly ascending vector, from *-1* to *1*. For this particular choice, $(DX)_j$ is simply the slope of the linear regression of the values in the jth row of D.

One exemplary choice for the weight matrix of *Neighborhood* is taken to be Gaussian $W_{ij} = e^{\frac{(i-j)^2}{2\sigma^2}}\cdot$ which is then normalized to be doubly stochastic (i.e. sum of each row and column is equal to one). For a given data set, there exists a range of relevant length scales, where large scales reflect the over all layout of the data, while smaller values give a better local organization at the expense of possibly fragmenting larger structures. This is captured in *SPIN* by controlling the value ofσ. One heuristic scheme that usually works well is starting with a very large sigma, iterating several times, then lowering σ (e.g. by a factor of *2*) and so forth.

Section II – Illustrative Examples and Applications

This Section describes some illustrative, non-limiting examples and applications for the method according to the present invention, demonstrated according to a preferred embodiment of the method, termed herein *SPIN*. A sorting algorithm, such as the method presented herein, is particularly useful in cases where the effect of some continuous parameter needs to be studied.

A specific example of the type of data where this form of analysis may be pertinent is biological experiments, such as genome-wide experiments for example. For example, the expression profile of synchronized cells is governed by the time in cell-cycle progression in which a particular sample was harvested, as demonstrated in Example 3. In Example 4, initial findings from the analysis of cancer data are presented. Example 5 demonstrates the use of the present invention for machine or robot vision.

In these cases, *SPIN*'s ability to ferret out elongated structures, even when the elongation refers to a complicated contour embedded in a high dimensional space, is extremely valuable.

Example 3

Yeast cell-cycle

A sorting algorithm, such as the one we present, is particularly useful in cases where the effect of some continuous parameter needs to be studied. A specific example of the type of data where this form of analysis may be pertinent is genome-wide experiments. For example, the expression profile of synchronized cells is governed by the time in cell-cycle progression in which a particular sample was harvested. In these cases, SPIN's ability to ferret out elongated structures, even when the elongation refers to a complicated contour embedded in a high dimensional space, is extremely valuable.

We chose to present here analysis of the yeast Elutriation-Synchronized cell-cycle expression data (taken from [1]). Spellman et al. employed a supervised 'phasing' method to assign genes to five known classes, namely G1, S, S/G2, G2/M and M/G1, utilizing the expression profiles of genes that were previously known to participate in specific phases of the cell cycle. They then proceeded to perform unsupervised analysis, specifically hierarchical clustering, and found that most genes belonging to the same class were clustered together. In another work, [2] further improved the organization of the tree by employing a leaf ordering algorithm, and recovered the order of the phases in the cycle.

Here we suggest the sorting approach as a different exploratory analysis methodology. Instead of partitioning the genes into distinct clusters we generate a distance matrix and order it by SPIN. As explained in Section I above, the nature of a cyclic object can be deduced from the colored pattern in the sorted distance matrix (fig. 9b): a blue elongated patch around the main diagonal, and two additional blue corners (upper right and lower left). Indeed, assigning such a cyclic nature to genes associated with cell-cycle is in accordance with known biological dynamics and functions [3]. Inspecting the sorted image at a higher resolution reveals the heterogeneous nature of the ring, indicating a further separation into the individual stages of the cycle. Therefore, information about the distinctive cell-cycle phases is not lost; while an understanding of the over all cyclic nature is gained.

The technical details of our analysis for this data set are as follows: The raw expression data was downloaded from a server at Stanford (http://cellcycle-www.stanford.edu), and included a total of 5,981 genes measured across 14 samples (which denote several consecutive stages along the cell-cycle). The only pre-processing step was a variance filter: the standard deviation was calculated for each of the 5,981 genes, and only the 600 genes with the highest values were chosen for analysis in SPIN. The gene distance matrix of size 600X600 was calculated using simply Euclidian distance metric, and sorted in SPIN, as shown in figure 9b. In this case the Neighborhood sorting algorithm was utilized, with $\sigma^2 = 5*600$ for 10 iterations. This example highlights the ease of ordering gene expression data in SPIN, and the informative and intuitive nature of the color-enhanced output provided by our tool. Previous studies have recognized the inherent cyclic nature of this data set [3], but required several stages of data manipulation and normalization, followed by a manual ordering along the PCA projection to convey the results that are easily captured in SPIN.

Figures 9A-9C show the following with regard to the analysis of yeast data according to the present invention: (a) The sorted expression matrix. The genes were sorted by *SPIN* and the matrix is ordered according to that permutation. The samples are ordered according to time. (b) The sorted distance matrix for the 600 genes, calculated in sample-space. The cyclic nature is quite visible. Upon closer inspection one can see that the ring separates into *3* main elongated clusters. (c) The projection of genes on the first and second PCA. Annotation of cell-cycle stages is based on the ordering presented in [3], to which *SPIN's* ordering has *85%* correlation.

In the general context of expression data SPIN provides a two-way sorting platform, i.e. it is possible to order both samples and genes. In the specific case of the yeast cell-cycle data the samples are already organized and labeled according to the stage in cell-cycle progression from which they were harvested. Therefore, we proceeded to sort only the genes, and left the samples ordered according to their labels. However, we did examine the organization of the Euclidian distance matrix for the samples (of size 14X14). One interesting observation is that the samples also order in a cyclic conformation of a ring. This observation is in accordance with the biology of the experiment, since each sample represents the expression profile of a yeast cell during consecutive stages of the cell-cycle.

References for Yeast section

1. Spellman PT, S.G., Zhang MQ, Iyer VR, Anders K, Eisen MB, Brown PO, Botstein D, Futcher B., Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. Mol Biol Cell, 1998. 9(12): p. 3273-3297.

2. Bar-Joseph Z, G.D., Jaakkola TS, Fast optimal leaf ordering for hierarchical clustering. Bioinformatics, 2001. 17: p. S22-9.

3. O. Alter, P.O.B.a.D.B., Singular Value Decomposition for Genome-Wide Expression Data Processing and Modeling. PNAS, 2000. 97(18): p. 10101-10106.

Example 4

Cancer research

The present invention used *SPIN* in the analysis of expression data originating from large-scale cancer experiments. A known problem in microarray experiments is that a given sample is usually contaminated by a mixture of cell types, so that the expression signal from the desired target may be partially masked [2].

The method of the present invention was used to analyze genomic data obtained from human leukemia patients. Expression data was used from Table 3 in ([17]), who identified 80 genes that separate Pro B-cell from pre B- and T-cell ALLs. The analysis presented in (10) may link those genes to hematopoiesis, which is the process of generation and differentiation of blood cells. During hematopoiesis stem cells divide and undergo differentiation to various stages, gradually losing their multipotencity (11).

In the present analysis both the samples and genes were reordered using SPIN. By looking at the reordered distance matrices in Figure 10 one can clearly see that the samples form an elongated cluster, which may be indicative of a gradual differentiation process. As shown in Figure 10 (sorted expression) the figure parts are as follows: Clockwise from top left: PCA of genes in sample-space, ordered distance matrix of genes, ordered expression matrix in logarithmic scale, distance matrix of the samples, PCA of the samples in gene-space.

When the expression data is ordered in both directions it becomes apparent that most (about 60) genes are gradually turning off, and a minority of 20 genes, specific of the final target of the process, are turned on as differentiation proceeds. The gradual decrease in transcription viewed here is in accordance with the hypothesis that stem cells possess an open chromatin structure, which is progressively quenched during differentiation (12).

EXAMPLE 5

MACHINE VISION

As previously described, the present invention is useful for any type of analysis problem involving the analysis of large sets of multidimensional data, including those characterized by having continuous variables. One example of such data is pattern recognition for machine or robot vision.

As an exemplary data set, the multi-feature digit dataset was examined. This dataset consists of features of handwritten numerals (`0'--`9') extracted from a collection of Dutch utility maps (13-14). Two hundred patterns per class (for a total of 2,000 patterns) have been digitized in binary images, which have subsequently been averaged in windows of 2x3, resulting in 240 averaged pixels per image. Each pattern is thus represented as a vector of 240 elements, with values ranging from 0 to 1. The 2000x2000 Euclidean distance matrix between the patterns was calculated; optionally other distance matrices could also be used. One advantage to selecting a simpler distance measure such as Euclidean distance is that the characteristics of the measure itself do not bias the results in a particular direction. The distance matrix was then sorted by *Neighborhood,* using a series of decreasing values of σ (σ^2 = 1,000,000, 500,000, 100,000, etc.) until convergence.

Figure 11 shows the results of analyzing the patterns required to recognition the numerals as numbers: (a) Dendrogram generated by the Ward linkage algorithm, hand colored to best represent the correct classification of the digits, identified by the blue dots on the bottom panel. This was done as a control in order to show that in the context of classifying, the quality of our method is as at least as good as common clustering methods. However, as the results show, the method according the present invention is better than common clustering methods, which are less successful for this type of problem. (b) The SPIN reordered distance matrix. As can be seen in the bottom panel the classification according to digit type is quite good. Moreover, from this distance matrix, several other features of the data become apparent. For example the digits as a whole seem to form an elongated structure with the digits ordered as follows: 4, 6, 0, 8, 5, 3, 1, 9, 7, 2. Some digits, such as 4 and 6 are very similar and seem to morph from one to the other, but are very different from other digits such as 7. (c) Looking in more detail at the sub-matrix of fours and sixes: First two PCA colored according to the order suggested by SPIN going from dark blue to dark red. A few sample digits are displayed in the appropriate locations. Note the growth of the "leg" of the "4" then the swing of the left-upward stroke from vertical to 45° then shrinkage of the leg, transformation into "6" and finally return of the upward stroke to vertical. (d)

The distance sub-matrix displaying an 'X' pattern. (e) Some sample digits arranged in the order of SPIN.

As can be seen in Figure 11, SPIN groups the images in good accordance with the known labels. A comparably accurate partition is also provided by hierarchical clustering, and presented in the form of a dendrogram. From the sorted distance matrix one can deduce further information: The overall layout of the data has an elongated shape, implying that the images lie along a complex trajectory, with one numeral morphing into the next. In fact if the sorted images are displayed consecutively, the resulting movie is relatively smooth, and the shapes appear to be gradually evolving over time. Even the transition between different classes is mostly gradual, as exemplified in the zoom-in where the left vertical stroke of the "4" tilts to the right, then the "4" morphs into "6" and the stroke tilts back towards the vertical. The relationship between the "4" and "6" clusters is of the type we termed "anti-parallel rods", as can be seen in Fig. 6, where the 2D projection of these points is presented. The hallmark of such a structure is an X shape on the distance matrix.

A zoom-in operation in this context refers to extracting a sub-matrix from the input data and regarding it as a "new" data-set. The distances are recalculated using only the remaining information in this sub-matrix. This is somewhat reminiscent of local PCA. SPIN thus allows the evaluation of the effects of sub-sets of the features on the data points. At the same time it also allows the evaluation of sub-sets of the data on the importance of the features. As an example, some of the pixels in the images of digits are always black (0) and thus do not contribute at all to the distances between patterns. Other pixels only change within a sub-set of the digits, and are thus important for discriminating between them, but not between others.

In this work we presented several data sets where the ordered distance matrix generated by *SPIN* was extremely helpful in uncovering the structure of the data. One of the examples demonstrating *SPIN*'s ability to reveal the layout of the data is the yeast cell-cycle. This data set was previously analyzed using hierarchical clustering [7]. Despite being a very useful visualization tool, hierarchical dendrograms do not give a clear indication of the relative positions, symmetries and shapes of the clusters. Another drawback of hierarchical clustering is the large number of possible leaf orderings of the clustering tree. The algorithm in [8] finds the optimal leaf-ordering with respect to the nearest-neighbors energy function, given a particular dendrogram. This energy function is a special case of *Neighborhood* with $W_{ij} = 1 |_{|i-j|=1}$. Moreover, the requirement of an ordering satisfying a given dendrogram could be too restrictive, especially since different clustering algorithms may give different results for the same data set. *SPIN*, on the other hand, provides an ordering of the objects using only the information available from the distance matrix, thus maintaining the ability to explore the entire permutation space, bypassing the need for a middle-man. Having said that, it may be beneficial to combine our sorting strategy with clustering. In such synergy the clustering algorithm would enhance the separation into clear clusters, while the sorter would help elucidate the shapes and relationships between the clusters.

Although *SPIN* can be viewed as a special case of dimensionality reduction (to one dimension), the emphasis is on ordering the points, rather than preserving their distances. Dimensionality reduction techniques, such as MDS, LLE [12] or Isomap [13], distort the distances. Therefore, the existence of a low-dimensional object can be discovered, however its structure is not readily inferred. Using *SPIN*, we have demonstrated that the re-ordered distance matrix highlights structural features of the object embedded in the high-dimensional space. Furthermore, we have also shown how *SPIN* can enhance dimensionality reduction techniques, as exemplified above where the color coded ordering significantly clarifies the *PCA* image. To conclude, the sole input to *SPIN* is a distance matrix (not necessarily Euclidian) which makes it applicable to any problem involving arrangements of points in multi-dimensional space, where a metric can be defined.

References

[1] M. Eisen, P.Spellman, P. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. Proc. Natl. Acad. Sci. USA, 95:14863–14868, 1998.

[2] U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proc. Natl. Acad. Sci. USA, 96:6745–6750, 1999.

[3] D. Tsafrir, W. Liu, Y. Yamaguchi, I. Tsafrir, Y. Wen, W. Gerald, R. Stengel, F. Barany, P. Paty, E. Domany, and D. Notterman. A novel mathematical approach to analyzing gene expression data: results from an international colon cancer consortium. In proc. of AACR 2004.

[4] T. Koopmans and M. Beckmann. Assignment problems and the location of economic activities. Econometrica, 25:53–76, 1957.

[5] R.E. Burkard, E. Cela, P. Pardalos, and S.L. Pitsoulis. The Quadratic Assignment Problem, volume 3, pages 241–339. Dordecht:Kluwer Academic Publishers, 1998.

[6] E. A. Dinic and M. A. Kronrod. An algorithm for the solution of the assignment problem. Soviet Math. Dokl., I0:1324–1326, 1969.

[7] P.T. Spellman, G. Sherlock, MQ. Zhang, V.R. Iyer, K. Anders, M.B. Eisen, P.O. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. Molecular Biology of the Cell, 9:3273–3297, 1998.

[8] Z. Bar-Joseph, D. K. Gi®ord, and T. S. Jaakkola. Fast optimal leaf ordering for hierarchical clustering. Bioinformatics (Proceedings of ISMB 2001), 17:22–29, 2001.

[9] O. Alter, P.O. Brown, and D. Botstein. Singular value decomposition for genome-wide expression data processing and modeling. Proc. Natl. Acad. Sci. USA, 97:10101–10106, 2000.

[10] C. Rosty, I. Tsafrir, N. Stransky, D. Tsafrir, J.P. Thiery, F. Radvanyi, E. Domany, and X. Sastre. Characterization of gene expression profiles in cervical carcinoma. submitted to AACR 2004.

[11] D.A. Notterman, U. Alon, A.J. Sierk, and A.J. Levine. Transcriptional gene expression profiles of colorectal adenoma, adenocarcinoma, and normal tissue examined by oligonucleotide arrays. Cancer Res., 61:3124–3130, 2001.

[12] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. Science, 290 (5500):2323–2326, 2000.

[13] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. Science, 290(5500):2319–2323, 2000.

[14] M. R. Garey and D. S. Johnson. Computer & Intractability: A Guide to the Theory of NP-Completeness. W H Freeman, 1979.

[15] C. H. Papadimitriou. The euclidean traveling salesman problem is np-complete. Theoretical Computer Science, 4(3):237–244, 1977.

[16] I. Tsafrir, L. Ein-Dor, O. Zuk, D. Tsafrir, and E. Domany. Iterative sorting algorithm for multidimentional data organization and visualization. in preperation, 2004.

[17]. Rozovskaia, T., Ravid-Amir, O., Tillib, S., Getz, G., Feinstein, E., Agrawal, H., Nagler, A., Rappeport, E. Issaeva, I., Matsuo, Y., Kees, U. R., Lapidot, T., Lo Coco, F., Foa, R., Mazo, A., Nakamura, T., Croce, C.M., Cimino, G., Domany, E. and Canaani, E, *PNAS* **100**, 7853 (2003).

Although the invention has been described in conjunction with specific embodiments thereof, it is evident that many alternatives, modifications and variations will be apparent to those skilled in the art. Accordingly, it is intended to embrace all such alternatives, modifications and variations that fall within the spirit and broad scope of the appended claims.

WHAT IS CLAIMED IS:

1. A method for analyzing data, comprising performing an unsupervised analysis of data according to a reordered distance matrix.

2. The method of claim 1, wherein said distance matrix is reordered using a weighting function.

3. The method of claims 1 or 2, suitable for automatically and semi-automatically analyzing data.

4. The method of any of claims 1-3, wherein the data comprises a plurality of objects characterized by continuous variables.

5. The method of any of claims 1-4, further comprising:

visualization of the data according to said analysis.

6. The method of claim 5, further comprising:

detecting at least one characteristic of the data according to said visualization.

7. The method of any of claims 1-6, further comprising:

detecting at least one characteristic of the data according to said analysis.

8. The method of claims 6 or 7, wherein the data is analyzed without reference to a predetermined order and/or wherein the data lacks pre-ordering.

9. The method of any of claims 1-8, comprising the SPIN method.

10. The method of claim 9, wherein the SPIN method comprises the *Side-to-Side (STS)* method, featuring a strictly increasing or decreasing vector for reordering said distance matrix.

11. The method of claim 10, wherein said STS method comprises:

Input: D_{nxn} and a strictly increasing vector X

1. Compute $S = D X$.

2. Sort S in descending order to get $S' = P(S)$, where P is the sorting permutation.

3. If $P(S) \mathrel{!}= S$, set $D = P D P^T$ and go to *stage 1.*

4. Output D.

12. The method of claim 11, further comprising performing stages 1-3 more than once.

13. The method of claims 11 or 12, further comprising using at least one heuristic to reorder *D*.

14. The method of claim 9, wherein the SPIN method comprises the *Neighborhood* method, featuring a matrix of fixed size.

15. The method of claim 14, wherein said Neighborhood method comprises:

```
Input : $D_{nxn}$ and $W_{nxn}$
1. Compute $M = D W$
2. Set $\mathrm{P} = \arg\min_{\mathrm{Q} \in \mathrm{S_n}} \mathrm{tr}(\mathrm{QM})$·
3. If tr ($P M$) != tr ($M$), set $D = P D PT$ and go to 1.
4. Output $D$.
```

16. The method of claim 15, further comprising performing stages 1-3 more than once.

17. The method of claims 15 or 16, further comprising using at least one heuristic to reorder *D*.

18. The method of claim 14, wherein the *Neighborhood* method features Gaussian smoothing.

19. The method of any of claims 15-18, wherein stage 2 is performed by solving the Linear Assignment Problem.

20. The method of any of claims 1-19, further comprising:

zooming in on a part of the data by separately examining a sub-matrix of the data according to said analysis.

21. The method of claim 20, further comprising:

separately examining a plurality of sub-matrices of the data according to said analysis; and

comparing results of said separate examinations to determine at least one characteristic of the data.

22. The method of any of claims 1 to 21, wherein the data comprises gene expression data and/or data from a gene microarray, comprising data from a large number of genes analyzed simultaneously.

23. The method of any of claims 1 to 21, wherein the data comprises data from expression of genes in cancerous tissue.

24. The method of any of claims 1 to 21, wherein the data comprises data related to a biological process, optionally including a biological cycle.

25. The method of any of claims 1 to 21 adapted for machine vision.

26. A method for analyzing gene expression data and/or data from a gene microarray, comprising data from a large number of genes analyzed simultaneously, comprising:

filtering the data according to a variance filter to form filtered data;

determining a distance matrix for said filtered data; and

reordering said distance matrix to analyze said filtered data.

27. The method of claim 26, further comprising:

analyzing said reordered distance matrix to determine at least one characteristic of said filtered data.

28. The method of claim 27, wherein said reordering is performed according to an automatic and/or semi-automatic, unsupervised analysis.

29. The method of claim 28, wherein said reordering is performed according to SPIN.

30. The method of any of claims 27-29, wherein the data is analyzed to determine a noise level in the data.

31. The method of claim 30, wherein said noise level is used to alter at least one characteristic of the microarray or of an experimental protocol for data collection.

32. The method of any of claims 27-29, wherein the data is analyzed to determine an inherent property of the data other than a property for which the experiment was designed.

33. The method of any of claims 26-32, wherein the data comprises cancer-related data.

34. The method of any of claims 26-33, adapted for ordering both samples and genes.

35. A method for analyzing data related to a biological process, optionally including a biological cycle, comprising the SPIN method.

36. A method for machine vision, comprising the SPIN method.

37. The method of claim 36, wherein the SPIN method is performed for analyzing a distance matrix for visual data.

38. The method of claim 37, further comprising:

zooming in on a part of the data by separately examining a sub-matrix of the data according to said analysis.

39. The method of claim 38, further comprising:

separately examining a plurality of sub-matrices of the data according to said analysis; and

comparing results of said separate examinations to determine at least one characteristic of the data.

40. A method according to any of claims 1-39, for partitioning the data into a plurality of optionally overlapping subsets.

41. The method of claim 40, further comprising:

using the distance matrices calculated from each subset separately to find novel partitions.

42. The method of any of claims 1-41, further comprising implementing the method and presenting the data with an intuitive easy-to-use GUI.

43. A method for analyzing data from expression of genes in cancerous tissue, comprising the SPIN method.

44. The method of any of claims 1-43, further comprising optionally constraining said reordering according to a dendrogram from any hierarchical clustering method.

ABSTRACT

A method for an unsupervised analysis of data according to a reordered distance matrix. According to preferred embodiments thereof, the present invention is useful for large scale multidimensional data, more preferably data having at least four dimensions. The present invention is also preferably used for data comprising a plurality of objects characterized by continuous variables, for example variables having a continuum of possible values rather than a plurality of discrete values.

Figure 1

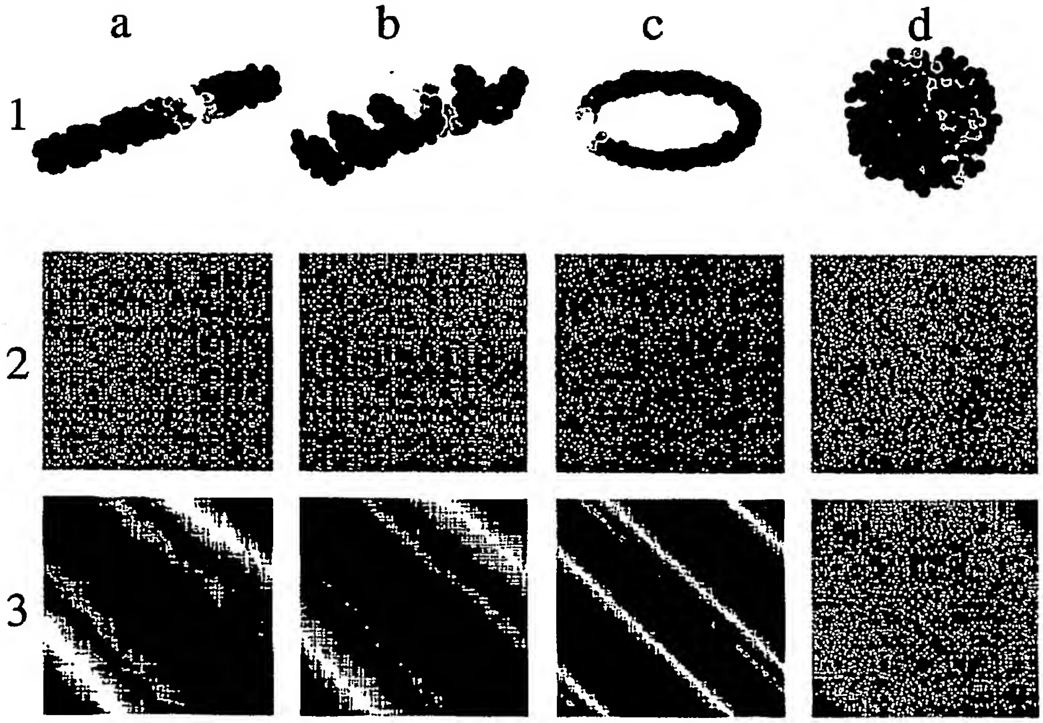


Figure 2

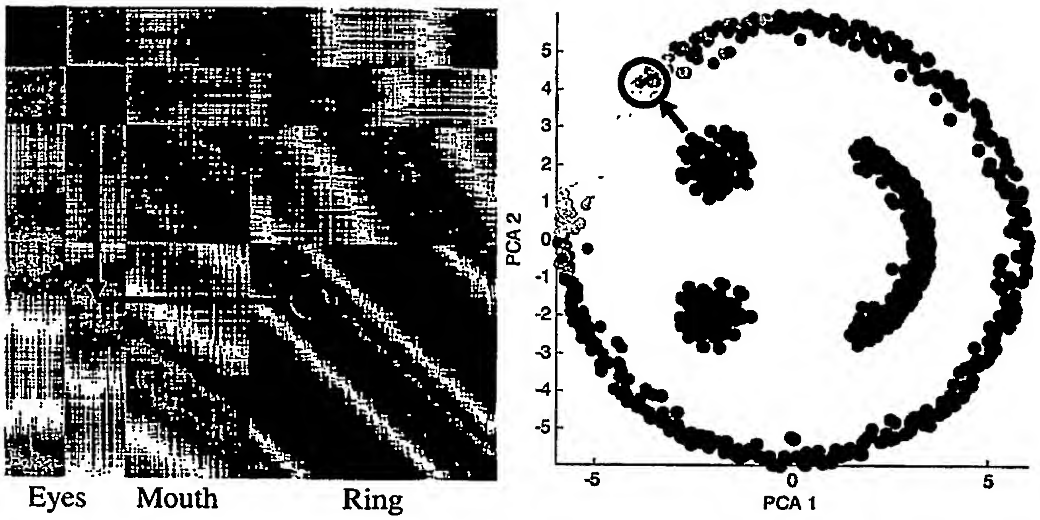


Figure 3

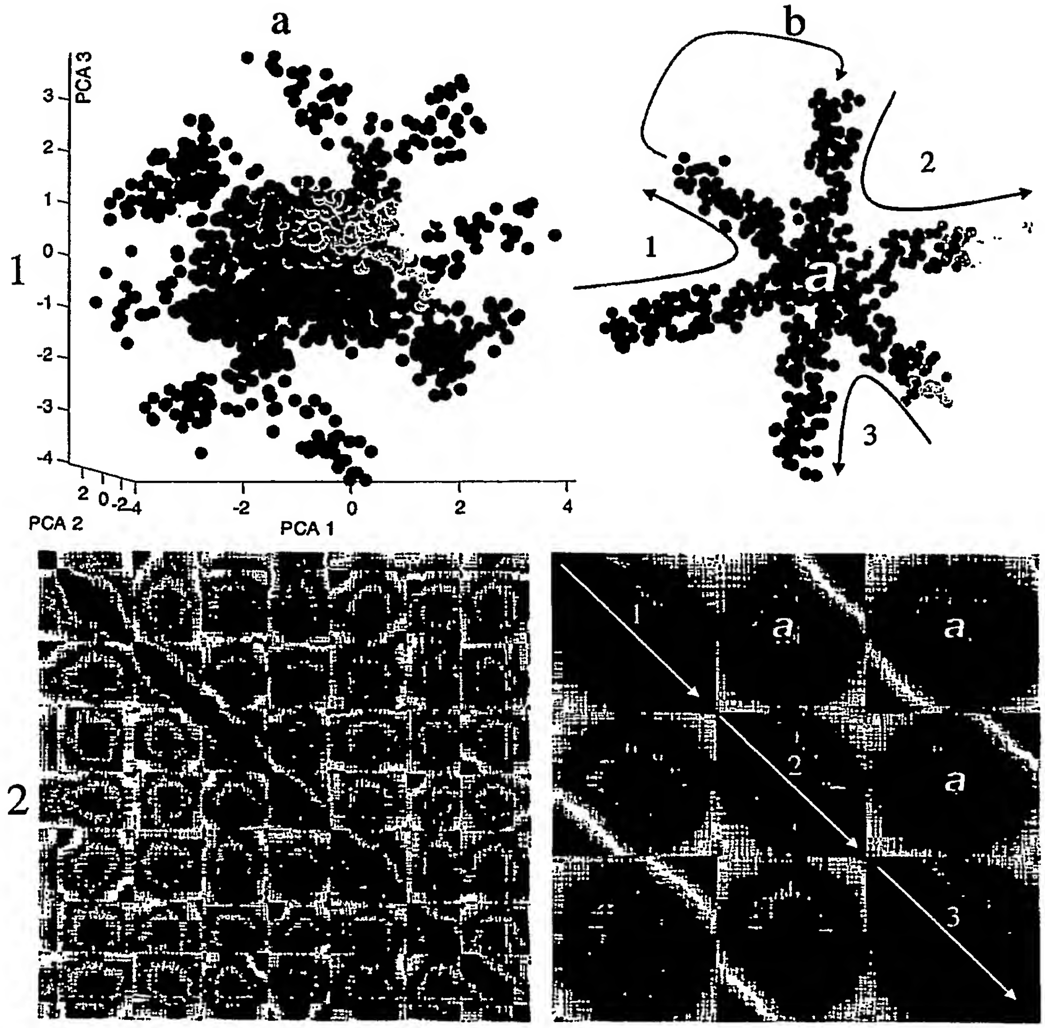
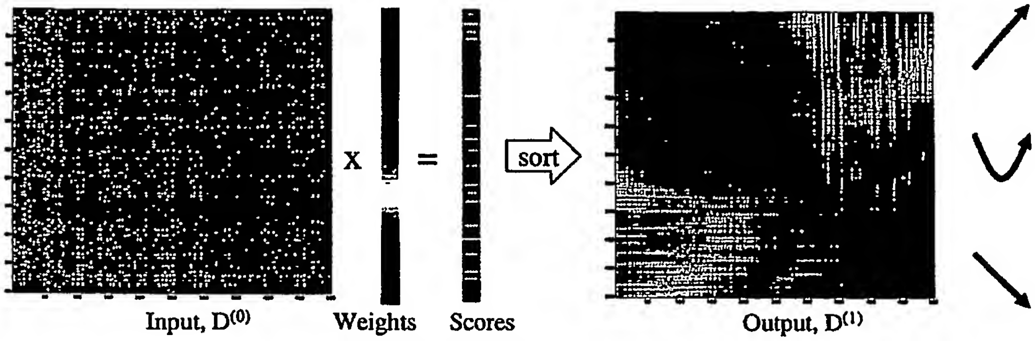


Figure 4



Schematic representation of Side-to-side sorting.

The initial unsorted distance matrix on the left is multiplied by a weight vector, resulting in a vector of scores. The weight vector's components increase linearly from -1 to 1. The points are then sorted according to their scores, generating a new permutation of the distance matrix, as shown on the right. This process is iterated until convergence, and the final outcome is shown at the right. By viewing the sorted matrix on the right it is readily apparent that the overall trend of the values in the upper rows is ascending, while the bottom rows have descending values, and intermediate rows have in-between values.

Figure 5 Side-by-Side

1. Define the weight vector $W_j = \frac{2j-N-1}{N-1}$
2. Calculate the scores vector $S_i^{(t)} = \sum_j D_{ij}^{(t)} W_j$
3. Sort the scores {k} = index sort({Si})
4. Reorder the distance matrix $D^{(t+1)} = D^{(t)}(\{k\},\{k\})$
5. Repeat steps 2-4 until $D^{(t+1)}$ is equal to $D^{(t)}$

Figure 6 – Results of Side-by-Side algorithm

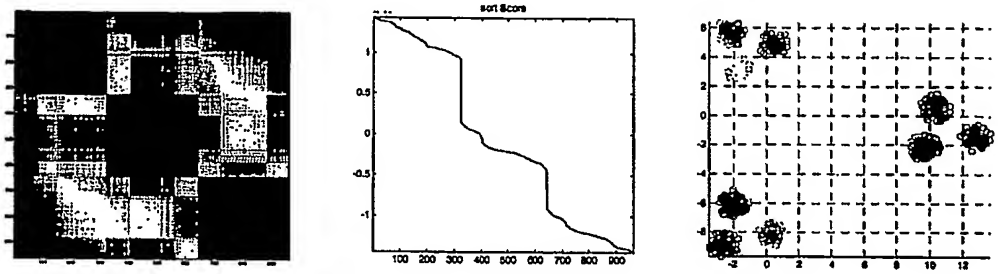
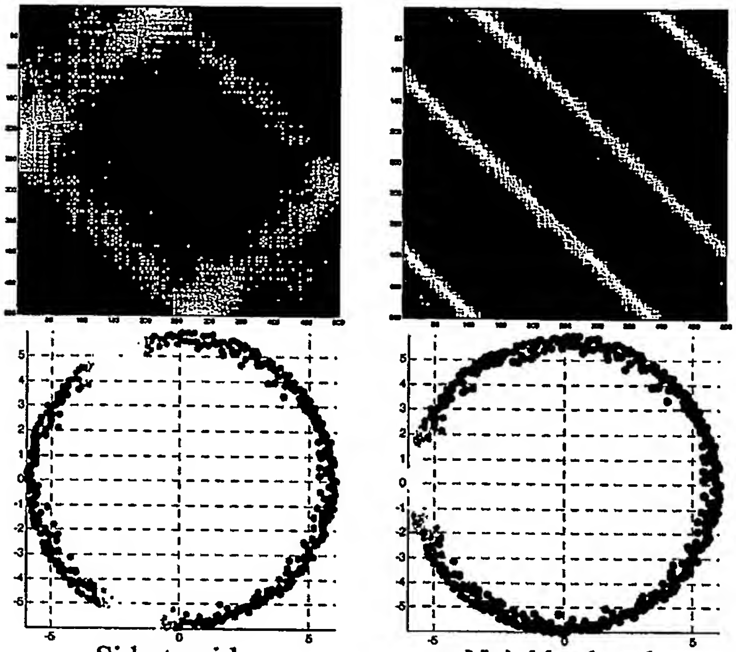


Figure 7 – Neighborhood algorithm

```
1. Define the weight matrix $W_{ij} = e^{\frac{-(i-j)^2}{\varepsilon \cdot N}} \Big/ \sum_k e^{\frac{-(k-j)^2}{\varepsilon \cdot N}}$
2. Calculate the mismatch matrix $M_{ij}^{(t)} = \sum_k D_{ik}^{(t)} W_{kj}$
3. Extract score vector $\mathrm{S_i} = \arg\min_j(\mathrm{M_{ij}})$
4. Sort the scores $\{k\}$ = index sort($\{Si\}$)
5. Reorder the distance matrix $D^{(t+1)} = D^{(t)}(\{k\},\{k\})$
6. Repeat steps 1-5 while adjusting $\varepsilon$
```

Figure 8



Side-to-side

Neighborhood

Figure 9

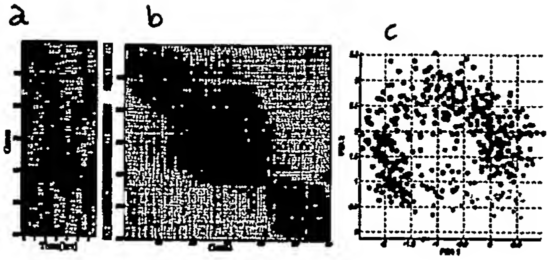


Figure 10

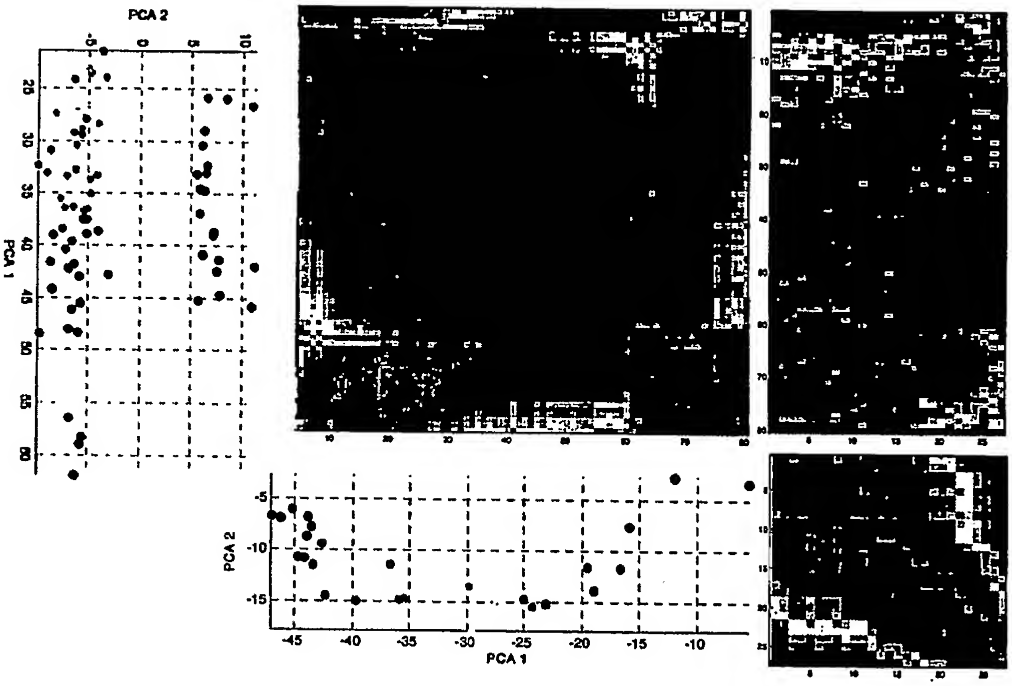
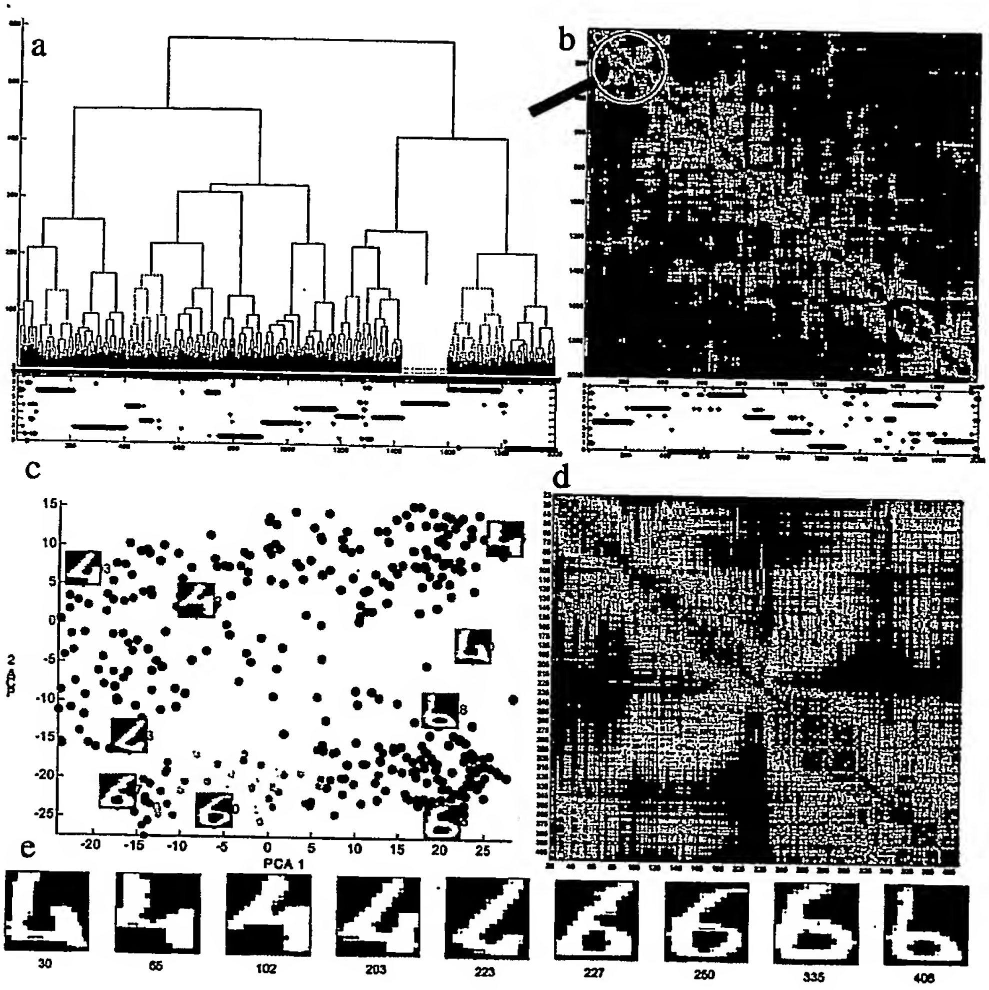

PCA 2
-5
0
5
10
25
30
35
40
45
50
55
60
PCA 1
10
20
30
40
50
60
70
80
-5
-10
-15
PCA 2
-45
-40
-35
-30
-25
-20
-15
-10
PCA 1

Figure 11


a
b
c
d
e
15
10
5
0
-5
-10
-15
-20
-25
PCA 2
-20
-15
-10
-5
0
5
10
15
20
25
PCA 1
30
65
102
203
223
227
250
335
408